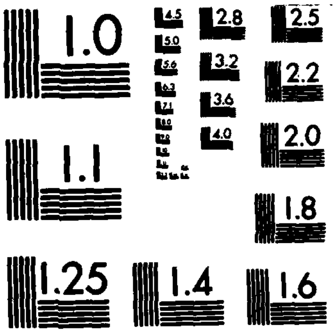
AD-A160 640 THIRD-ORDER CO-OCCURRENCE TEXTURE ANALYSIS APPLIED TO SAMPLES OF HIGH RES. (U) ARMY ENGINEER TOPOGRAPHIC LABS FORT BELVOIR VA R A HEVENOR AUG 85 ETL-0396 1/1

UNCLASSIFIED F/G 17/9 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

P

ETL – 0396

Third – order co – occurrence texture analysis applied to samples of high resolution synthetic aperture radar imagery

Richard A. Hevenor

August 1985

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

Prepared for
U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060 – 5546

E
T
L

US ARMY ENGINEER TOPOGRAPHIC LABORATORIES
ETL

Destroy this report when no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The citation in this report of trade names of commercially available products does not constitute official endorsement or approval of the use of such products.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL-0396	2. GOVT ACCESSION NO. AD-A160 640
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THIRD-ORDER CO-OCCURRENCE TEXTURE ANALYSIS APPLIED TO SAMPLES OF HIGH RESOLUTION SYNTHETIC APERTURE RADAR IMAGERY	5. TYPE OF REPORT & PERIOD COVERED Technical Report January 1984 - June 1984
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard A. Hevenor	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060-5546	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 4A161101A91D
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060-5546	12. REPORT DATE August 1985
	13. NUMBER OF PAGES 37
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Texture Analysis,
Pattern Recognition, and
Radar Imagery.

Keywords include

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A third-order co-occurrence texture analysis technique was applied to samples of synthetic aperture radar imagery. This technique was applied to four classes of terrain features on selected samples of radar imagery. The four classes considered were forests, cities, agricultural fields, and water. A feature vector was computed from samples of each class. A linear feature selection technique was applied for dimensionality reduction, and a pseudo-inverse classifier was developed. This classifier was applied to the original training samples as well as to other samples from the same imagery.

PREFACE

This study was conducted under ILIR Project 4A161101A91D, "Development of A New Technique for Texture Analysis."

The work described in this research note represents an application of a third-order co-occurrence approach toward texture analysis. The task was performed under the supervision of Dr. Frederick W. Rohde, Team Leader, Center for Physical Sciences; and Dr. Robert D. Leighty, Director, Research Institute.

COL Alan L. Laubscher, CE, was the Commander and Director and Mr. Walter E. Boge was the Technical Director of the Engineer Topographic Laboratories during the report preparation.

CONTENTS

ILLUSTRATIONS

TABLES

THIRD-ORDER CO-OCCURRENCE TEXTURE ANALYSIS APPLIED TO SAMPLES OF HIGH RESOLUTION RADAR IMAGERY

INTRODUCTION

The purpose of this research note is to show the application of a third-order co-occurrence texture analysis to samples of high resolution synthetic aperture radar imagery. In the past, various types of texture analysis techniques have been applied to digital images. One of the most popular techniques has been the computation of the gray level co-occurrence matrix. Each element of this matrix represents the number of times that gray level ℓ occurs next to gray level m for a given pixel spacing and direction. The concept employed in this report is a natural extension of this technique. In a third-order co-occurrence analysis, calculations are made for the number of times gray level ℓ occurs next to gray level m, which in turn occurs next to gray level n. An interesting aspect of this jump from second- to third-order co-occurrence is that now all three pixels do not necessarily have to lie in a straight line. The pixel spacing and direction between gray level ℓ and gray level m can be quite different from the spacing and direction between gray level m and gray level n. The following sections will present a discussion of the third-order co-occurrence technique along with the feature selection and classification methods applied to certain samples of radar imagery.

METHODOLOGY

This section develops the third-order co-occurrence method and discusses briefly the methods used for feature selection and classification. The third-order approach computes the number of times in a given image that gray level ℓ occurs next to gray level m, which in turn occurs next to gray level n. The pixel spacing between gray level ℓ and gray level m will be denoted by Δx_1 and Δy_1, and the spacing between gray level ℓ and gray level n will be denoted by Δx_2 and Δy_2. The functional notation for the third-order co-occurrence is

$$N(i, j, k|\Delta x_1, \Delta y_1, \Delta x_2, \Delta y_2)$$

where $i=\ell+1$ $j=m+1$ $k=n+1$

The above expression will be shortened to N_{ijk}, where the values for the four spacing parameters are assumed. N_{ijk} can be visualized as a third-order array, the dimensions of which will be determined by the total number of gray levels in the digital image. If the number of gray levels in the image is 16, then N_{ijk} is an array whose dimensions are 16 by 16 by 16. Various measures or features can then be computed from N_{ijk} to form a feature vector that can be used for classification purposes. The 13 features given below are calculated from N_{ijk} and used to form the components of a feature vector X.

$$\text{Second Moment} = \frac{\sum_i \sum_j \sum_k N_{ijk}^2}{\sum_i \sum_j \sum_k N_{ijk}} = x_1 \qquad (1)$$

$$\text{Small Number Emphasis} = \frac{\sum_i \sum_j \sum_k N_{ijk} \;/\; (i^2 + j^2 + k^2)}{\sum_i \sum_j \sum_k N_{ijk}} = x_2 \qquad (2)$$

$$\text{Large Number Emphasis} = \frac{\sum_i \sum_j \sum_k (i^2 + j^2 + k^2)\, N_{ijk}}{\sum_i \sum_j \sum_k N_{ijk}} = x_3 \qquad (3)$$

$$\text{Depth Emphasis} = \frac{\sum_i \sum_j \left[\sum_k N_{ijk}\right]^2}{\sum_i \sum_j \sum_k N_{ijk}} = x_4 \qquad (4)$$

$$\text{Row Emphasis} = \frac{\sum_j \sum_k \left[\sum_i N_{ijk}\right]^2}{\sum_i \sum_j \sum_k N_{ijk}} = x_5 \qquad (5)$$

$$\text{Column Emphasis} = \frac{\sum_i \sum_k \left[\sum_j N_{ijk}\right]^2}{\sum_i \sum_j \sum_k N_{ijk}} = x_6 \qquad (6)$$

$$\text{Contrast} = \sum_i \sum_j \sum_k \left[(i-j)^2 + (j-k)^2 + (i-k)^2\right] N_{ijk} = x_7 \qquad (7)$$

$$\text{Entropy} = \frac{\sum_i \sum_j \sum_k N_{ijk} \log N_{ijk}}{\sum_i \sum_j \sum_k N_{ijk}} = x_8 \tag{8}$$

$$\text{Third Moment} = \frac{\sum_i \sum_j \sum_k \left[N_{ijk} \right]^3}{\sum_i \sum_j \sum_k N_{ijk}} = x_9 \tag{9}$$

$$\text{Absolute Value} = \frac{\sum_i \sum_j \sum_k \left[|i-j| + |j-k| + |i-k| \right] N_{ijk}}{\sum_i \sum_j \sum_k N_{ijk}} = x_{10} \tag{10}$$

A general expression for the summation of terms along specific diagonals is given by A_{ab} below:

$$A_{ab} = \sum_{\substack{i \\ |i-j|=a}} \sum_{j} \sum_{\substack{k \\ |i-k|=b}} N_{ijk}$$

Using the above definition, the following three features can be computed:

$$A_{00} = \sum_{\substack{i \\ |i-j|=0}} \sum_{j} \sum_{\substack{k \\ |i-k|=0}} N_{ijk} = x_{11} \tag{11}$$

$$A_{01} = \sum_{\substack{i \\ |i-j|=0}} \sum_{j} \sum_{\substack{k \\ |i-k|=1}} N_{ijk} = x_{12} \tag{12}$$

$$A_{10} = \sum_{\substack{i \\ |i-j|=1}} \sum_{j} \sum_{\substack{k \\ |i-k|=0}} N_{ijk} = x_{13} \tag{13}$$

In all the above definitions, the summations in i, j, and k are over the entire range of gray levels in the image. If there are 16 gray levels in the image, then i, j, and k all vary from 1 to 16. There is nothing particularly unique about the 13 features chosen above. Other features could also be computed from N_{ijk} and perhaps would even work better than the set chosen. Once the form of the feature vector X has been set, a method for performing feature selection needs to be considered.

In this study the feature selection technique came from the field of discriminant analysis of statistics. The feature selection method was used to reduce the dimensionality of X from 13 to 2 and at the same time optimize the separation among data belonging to different classes. The feature selection operation can be performed by using a linear transformation as follows:

$$Y = AX \tag{14}$$

where X is the original feature vector with dimensionality 13x1; A is the transformation matrix of dimensionality 2x13; and Y is the transformed feature vector with the dimensionality 2x1. In order to calculate the matrix A, use was made of the within-class and between-class scatter matrices. A within-class scatter matrix shows the scatter of samples around their class expected vector and can be expressed by

$$S_w = \sum_{i=1}^{N_t} P(\omega_i)\, C_i \tag{15}$$

where S_w is the within-class scatter matrix, $P(w_i)$ is the a priori probability of the i^{th} class, C_i is the covariance matrix of the i^{th} class, and N_t is the total number of classes. A between-class scatter matrix can be defined in many ways, however, the following definition was the one utilized here:

$$S_b = C_1 + C_2 + (M_1 - M_2)\,(M_1 - M_2)^T \tag{16}$$

where S_b is the between-class scatter matrix; C_1 is the covariance matrix for class 1; C_2 is the covariance matrix for class 2; M_1 is the mean vector for class 1; M_2 is the mean vector for class 2; and T means transpose. This definition of the between-class scatter matrix is valid only for the case when N_t (the total number of classes) is equal to two. In order to have criteria for class separability, a number must be derived from the scatter matrices. This number should increase when the distances between points belonging to different classes are increasing or when the distances between points belonging to the same class are decreasing. One criterion is the use of J_1, which is defined as follows:

where

$$J_1 = \text{trace}\,(S_{2m}^{-1}\ S_{1m})$$

$$S_{2m} = AS_wA^T \text{ and } S_{1m} = AS_bA^T \qquad (17)$$

The feature selection problem now requires that we select the particular transformation matrix A, which maximizes the value of J_1. Fukunaga[1] shows that A is made up of the normalized eigenvectors of the matrix $S_w^{-1}\ S_b$.

$$A^T = [\,\phi_1 \quad \phi_2\,] \qquad (18)$$

where ϕ_1 is the eigenvector associated with the largest eigenvalue, and ϕ_2 is the eigenvector associated with the second largest eigenvalue. Once the matnx A is computed from (18), the new feature vector Y can be computed for each point in each class. After feature selection has been performed, it is necessary to choose a classification procedure.

The classification procedure used in this work was the pseudoinverse technique. This technique is a linear classifier, which attempts to solve for a vector $\underline{a}$ such that

$$\text{if} \quad \underline{a}^T\underline{y} > 0 \quad \text{then decide class 1}$$

or

$$\text{if} \quad \underline{a}^T\underline{y} < 0 \quad \text{then decide class 2}$$

where the vector $\underline{y}$ is an augmented form of Y as shown below:

$$\underline{y} = \begin{bmatrix} 1 \\ y_1 \\ y_2 \end{bmatrix}$$

where y_1 and y_2 are the components of Y. The above decision function is valid for the two class problem. It can easily be seen that the vector $\underline{a}$ for our case has three components.

[1] Keinosuke Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 1972.

A solution for $\underline{a}$ can be found by forming a matrix H from all the training samples taken from the two classes. Each row of H will consist of a sample of $\underline{y}^T$. The matrix H will therefore have the following form:

$$H = \begin{bmatrix} 1 & y_{11} & y_{21} \\ 1 & y_{12} & y_{22} \\ 1 & y_{13} & y_{23} \\ \vdots & \vdots & \vdots \\ 1 & y_{1n} & y_{2n} \\ -1 & -y_{1n+1} & -y_{2n+1} \\ -1 & -y_{1n+2} & -y_{2n+2} \\ -1 & -y_{1n+3} & -y_{2n+3} \\ \vdots & \vdots & \vdots \\ -1 & -y_{1n+m} & -y_{2n+m} \end{bmatrix} \qquad (19)$$

The first *n* rows of H consist of data that come from class 1. The next *m* rows of H come from class 2 and have all been multiplied by -1. The second subscript on the y values refers to the sample number. An equation involving H and $\underline{a}$ is given below:

$$H\underline{a} = \underline{b} \qquad (20)$$

The vector $\underline{b}$ has *n+m* components all of which are arbitrarily specified positive constants. Duda and Hart[2] show how the following solution for $\underline{a}$ comes from equation (20):

$$\underline{a} = (H^TH)^{-1}\, H^T\underline{b} \qquad (21)$$

Once the three components of $\underline{a}$ have been calculated for each possible pair of classes, then the classifier has been completed and can be tested against the original training set data and some unknown samples.

[2] Richard O. Duda and Peter E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, 1973.

INVESTIGATION

The third-order co-occurrence texture calculations were applied to samples of high resolution synthetic aperture radar imagery taken over the Elizabeth City, North Carolina area with the UPD-4 radar system. Sections of the radar imagery were digitized and stored on a digital disk unit. A Lexidata system 3400 display processor was used to display the images on a cathode ray tube and to take 700 samples from four terrain classes. Each sample consisted of a 32 by 32 pixel element window located within a homogeneous section of one particular terrain class. The four classes considered were forests, fields, cities, and water. Of the 700 samples taken, 200 came from forests, 200 from fields, 200 from water, and 100 from cities. A training set consisting of 100 samples from each of the four classes was used to compute the A matrices and the $\underline{a}$ vectors for each of the six possible pairs of classes. Before the matrix A and the vector $\underline{a}$ could be calculated, however, it was necessary to compute J_1 as a function of several values of Δx_1, Δy_1, Δx_2, and Δy_2 in an attempt to find the spacings that yield a significantly large value for J_1. A computer program was written for the Hewlett-Packard 1000 computer to calculate J_1 as a function of the spacing values. This program was also used to compute the transformation matrix A. A listing of this program along with the subroutine used to compute the third-order co-occurrence array is given in appendix A. A second computer program was written to calculate the new feature vector Y and to compute the $\underline{a}$ vector from the components of Y. A listing of this second program is provided in appendix B. A third computer program was written that takes image samples and makes a classification, using as input the A matrix and the $\underline{a}$ vector data calculated in the previous two programs. The listing for this third program is given in appendix C.

RESULTS

In this section some results of numerical calculations are presented. Table 1 shows the results of computing the value of J_1 for each of the six possible pairs of classes. The values for the spacing parameters (Δx_1, Δy_1, Δx_2, Δy_2) shown in table 1 are those that yielded a maximum value for J_1 for each pair of classes.

Table 1. Values of J_1 for Third-Order Features

	Δx_1	Δy_1	Δx_2	Δy_2	J_1
Forests and Fields	1	0	2	0	49.78
Forests and Cities	2	0	2	1	23.05
Cities and Fields	1	0	2	0	169.21
Forests and Water	1	0	2	0	561.63
Fields and Water	1	0	2	0	196.66
Cities and Water	1	0	2	0	2790.16

It can easily be seen that the largest three values of J_1 all occur with water and some other class. These three values of J_1 are large enough to say that the data from these respective classes are well separated and clustered. The smallest value of J_1 occurs for forests and cities and indicates that the classes are not completely separated. Using the spacing values given in table 1, the matrix A and vector $\underline{a}$ were computed for each of the six pairs of classes, as shown below:

1. Forests and Fields

$$A = \left[\begin{matrix} -2.3\mathrm{x}10^{-3} & -0.996 & -1.01\mathrm{x}10^{-4} & 2.64\mathrm{x}10^{-3} & 9.45\mathrm{x}10^{-5} & 3.86\mathrm{x}10^{-4} & 1.2\mathrm{x}10^{-5} \\ 3.94\mathrm{x}10^{-3} & 0.999 & 2.78\mathrm{x}10^{-5} & -1.01\mathrm{x}10^{-2} & 6.74\mathrm{x}10^{-3} & 1.68\mathrm{x}10^{-3} & 1.33\mathrm{x}10^{-6} \end{matrix}\right.$$

$$\left.\begin{matrix} 4.12\mathrm{x}10^{-2} & -3.89\mathrm{x}10^{-5} & -7.95\mathrm{x}10^{-2} & -2.07\mathrm{x}10^{-4} & -5.08\mathrm{x}10^{-4} & -4.74\mathrm{x}10^{-4} \\ -4.43\mathrm{x}10^{-2} & -8.15\mathrm{x}10^{-6} & -1.00\mathrm{x}10^{-2} & -9.40\mathrm{x}10^{-5} & -7.7\mathrm{x}10^{-5} & -2.28\mathrm{x}10^{-5} \end{matrix}\right]$$

$$\underline{a} = \begin{bmatrix} -5.63792 \\ -19.91341 \\ -0.00121 \end{bmatrix}$$

2. Forests and Cities

$$A = \left[\begin{matrix} 1.57\mathrm{x}10^{-2} & -0.993 & 3.5\mathrm{x}10^{-5} & -1.43\mathrm{x}10^{-3} & -2.07\mathrm{x}10^{-4} & 2.84\mathrm{x}10^{-3} & 2.21\mathrm{x}10^{-6} \\ 2.4\mathrm{x}10^{-3} & -0.992 & -4.34\mathrm{x}10^{-5} & 1.07\mathrm{x}10^{-2} & 1.35\mathrm{x}10^{-3} & -9.54\mathrm{x}10^{-3} & -1\mathrm{x}10^{-5} \end{matrix}\right.$$

$$\left.\begin{matrix} -0.117 & 2.57\mathrm{x}10^{-4} & -1.07\mathrm{x}10^{-2} & -5.79\mathrm{x}10^{-5} & 1.39\mathrm{x}10^{-4} & -8.72\mathrm{x}10^{-5} \\ 0.101 & -1.57\mathrm{x}10^{-4} & 7.33\mathrm{x}10^{-2} & 4.36\mathrm{x}10^{-4} & 2.64\mathrm{x}10^{-4} & 4.31\mathrm{x}10^{-4} \end{matrix}\right]$$

$$\underline{a} = \begin{bmatrix} 1.40889 \\ 113.96356 \\ -0.00306 \end{bmatrix}$$

3. Cities and Fields

$$A = \left[\begin{matrix} -4.93\mathrm{x}10^{-2} & -1.27\mathrm{x}10^{-1} & 3.6\mathrm{x}10^{-4} & 5.68\mathrm{x}10^{-3} & -6.98\mathrm{x}10^{-4} & 7.44\mathrm{x}10^{-3} & 2.53\mathrm{x}10^{-5} \\ -5.11\mathrm{x}10^{-3} & 9.98\mathrm{x}10^{-1} & 2.11\mathrm{x}10^{-5} & -1.46\mathrm{x}10^{-2} & 1.21\mathrm{x}10^{-2} & 3.28\mathrm{x}10^{-3} & 4.51\mathrm{x}10^{-6} \end{matrix} \right.$$

$$\left. \begin{matrix} -9.77\mathrm{x}10^{-1} & 1.07\mathrm{x}10^{-4} & -1.62\mathrm{x}10^{-1} & -3.22\mathrm{x}10^{-4} & -1.66\mathrm{x}10^{-3} & 3.86\mathrm{x}10^{-4} \\ -3.93\mathrm{x}10^{-2} & 3.6\mathrm{x}10^{-5} & -3.63\mathrm{x}10^{-2} & -1.45\mathrm{x}10^{-4} & -3.24\mathrm{x}10^{-4} & -1.29\mathrm{x}10^{-4} \end{matrix} \right]$$

$$\underline{a} = \begin{bmatrix} 0.44802 \\ -3.44841 \\ 0.00066 \end{bmatrix}$$

4. Forests and Water

$$A = \left[\begin{matrix} -1.99\mathrm{x}10^{-3} & -6.79\mathrm{x}10^{-1} & -3.32\mathrm{x}10^{-4} & 2.22\mathrm{x}10^{-4} & 3.74\mathrm{x}10^{-4} & 5.3\mathrm{x}10^{-4} & -5.7\mathrm{x}10^{-6} \\ -2.75\mathrm{x}10^{-2} & 7.77\mathrm{x}10^{-1} & 2.4\mathrm{x}10^{-4} & -7.63\mathrm{x}10^{-4} & 1.53\mathrm{x}10^{-2} & 1.52\mathrm{x}10^{-3} & 1.64\mathrm{x}10^{-6} \end{matrix} \right.$$

$$\left. \begin{matrix} -7.26\mathrm{x}10^{-1} & -4.9\mathrm{x}10^{-7} & 1.14\mathrm{x}10^{-1} & 1.9\mathrm{x}10^{-3} & -4.24\mathrm{x}10^{-3} & 1.61\mathrm{x}10^{-3} \\ 6.28\mathrm{x}10^{-1} & 1.35\mathrm{x}10^{-5} & -3.5\mathrm{x}10^{-2} & 2.04\mathrm{x}10^{-3} & -5.8\mathrm{x}10^{-4} & -8.2\mathrm{x}10^{-4} \end{matrix} \right]$$

$$\underline{a} = \begin{bmatrix} 1.87565 \\ -1.05647 \\ 0.00005 \end{bmatrix}$$

5. Fields and Water

$$A = \left[\begin{matrix} 9.93\mathrm{x}10^{-3} & -6.58\mathrm{x}10^{-1} & -2.98\mathrm{x}10^{-4} & -2.62\mathrm{x}10^{-3} & -3.99\mathrm{x}10^{-3} & 9.63\mathrm{x}10^{-4} & -4.34\mathrm{x}10^{-6} \\ 3.57\mathrm{x}10^{-2} & -4.08\mathrm{x}10^{-1} & -1.82\mathrm{x}10^{-4} & -1.45\mathrm{x}10^{-3} & -1.72\mathrm{x}10^{-2} & -1.58\mathrm{x}10^{-3} & 6.17\mathrm{x}10^{-6} \end{matrix} \right.$$

$$\left. \begin{matrix} -7.48 \times 10^{-1} & -5.76 \times 10^{-6} & 8.65 \times 10^{-2} & 9.88 \times 10^{-5} & 1.05 \times 10^{-3} & 2.54 \times 10^{-3} \\ -9.05 \times 10^{-1} & -1.84 \times 10^{-5} & -1.08 \times 10^{-1} & -4.13 \times 10^{-3} & -1.5 \times 10^{-3} & -2.72 \times 10^{-3} \end{matrix} \right]$$

$$\underline{a} = \begin{bmatrix} 3.90389 \\ -0.26522 \\ -0.00047 \end{bmatrix}$$

6. Cities and Water

$$A = \left[\begin{matrix} 5.94 \times 10^{-3} & 1.45 \times 10^{-1} & 2.27 \times 10^{-5} & -1.36 \times 10^{-3} & -1.23 \times 10^{-3} & -2.31 \times 10^{-4} & 2.01 \times 10^{-6} \\ 1.12 \times 10^{-2} & -9.56 \times 10^{-1} & -2.06 \times 10^{-4} & 3.14 \times 10^{-3} & -2.98 \times 10^{-3} & -8.57 \times 10^{-3} & 9.29 \times 10^{-7} \end{matrix} \right.$$

$$\left. \begin{matrix} 9.89 \times 10^{-1} & -2.0 \times 10^{-6} & -3.55 \times 10^{-2} & -9.18 \times 10^{-4} & 5.24 \times 10^{-3} & 6.68 \times 10^{-4} \\ -2.93 \times 10^{-1} & -5.01 \times 10^{-6} & -3.51 \times 10^{-3} & 7.05 \times 10^{-4} & -1.29 \times 10^{-3} & 4.25 \times 10^{-3} \end{matrix} \right]$$

$$\underline{a} = \begin{bmatrix} 1.31629 \\ 0.94784 \\ -0.00007 \end{bmatrix}$$

Figures 1 through 6 show plots of the training samples for the two components of Y. Figure 1 shows a plot of the Y data for forests and fields. The X's represent data from forest samples, and the squares represent fields. The line in the center is the decision boundary provided by the pseudoinverse classifier. It can be seen that the data for forests and fields are not completely separated since three points belonging to the class of forests are located to the right of the final decision boundary. Figure 2 shows a plot of the Y data for forests and cities. These two classes are not completely separated either as shown by the location of the decision boundary. Figure 3 shows a plot of the Y data for cities and fields. These two classes are well separated, which corresponds to the large value of J_1. Figures 4,5, and 6 depict the Y data for forests and water, fields and water, and cities and water. In each of these three cases the Y data are very well separated and clearly defined. Once the decision boundaries were computed for each of the six possible pairs of classes, the classifier was used to determine the class of each of the 400 training samples. The results of this test are given in table 2.

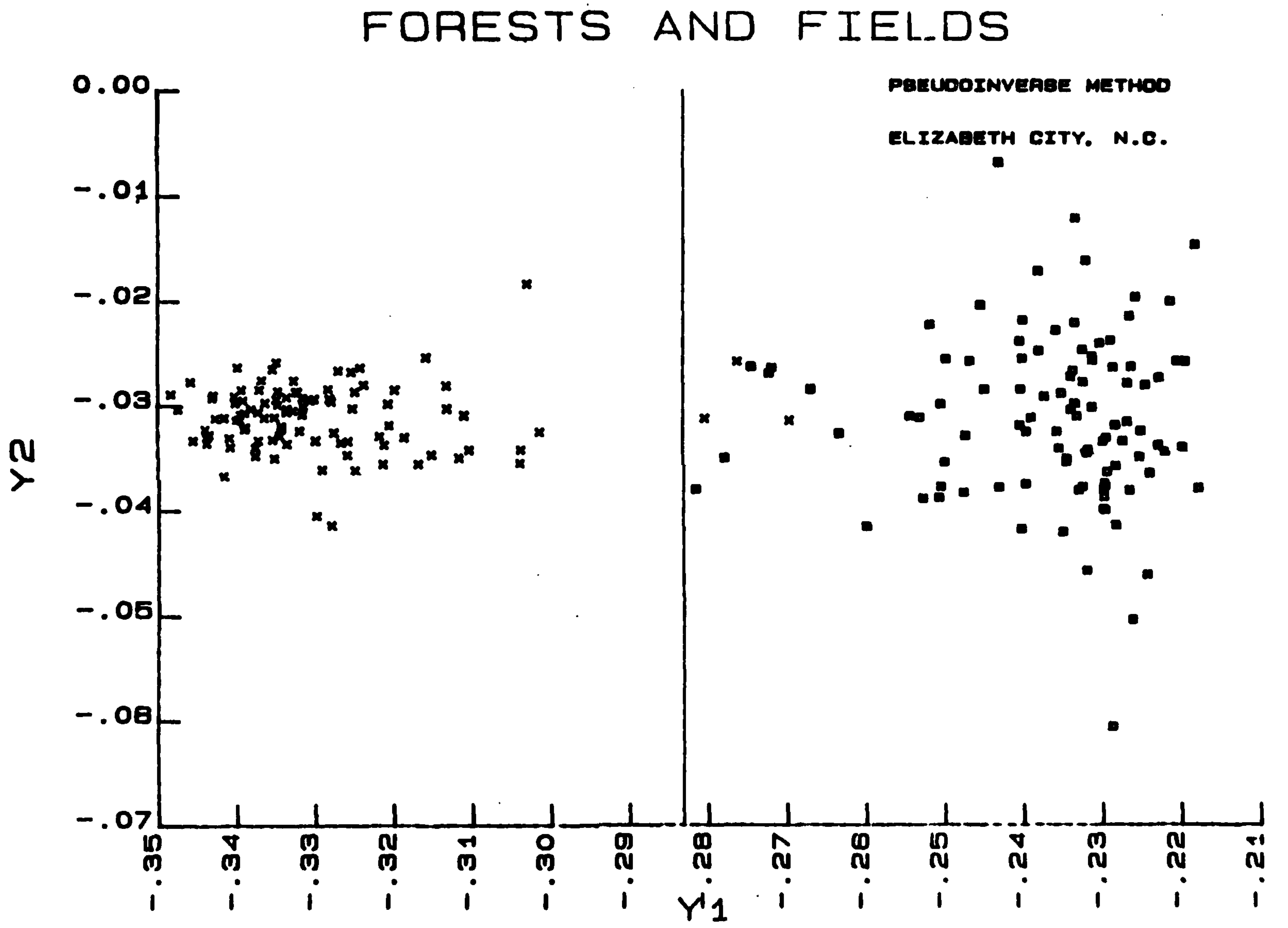


Figure 1. Forests and Fields.

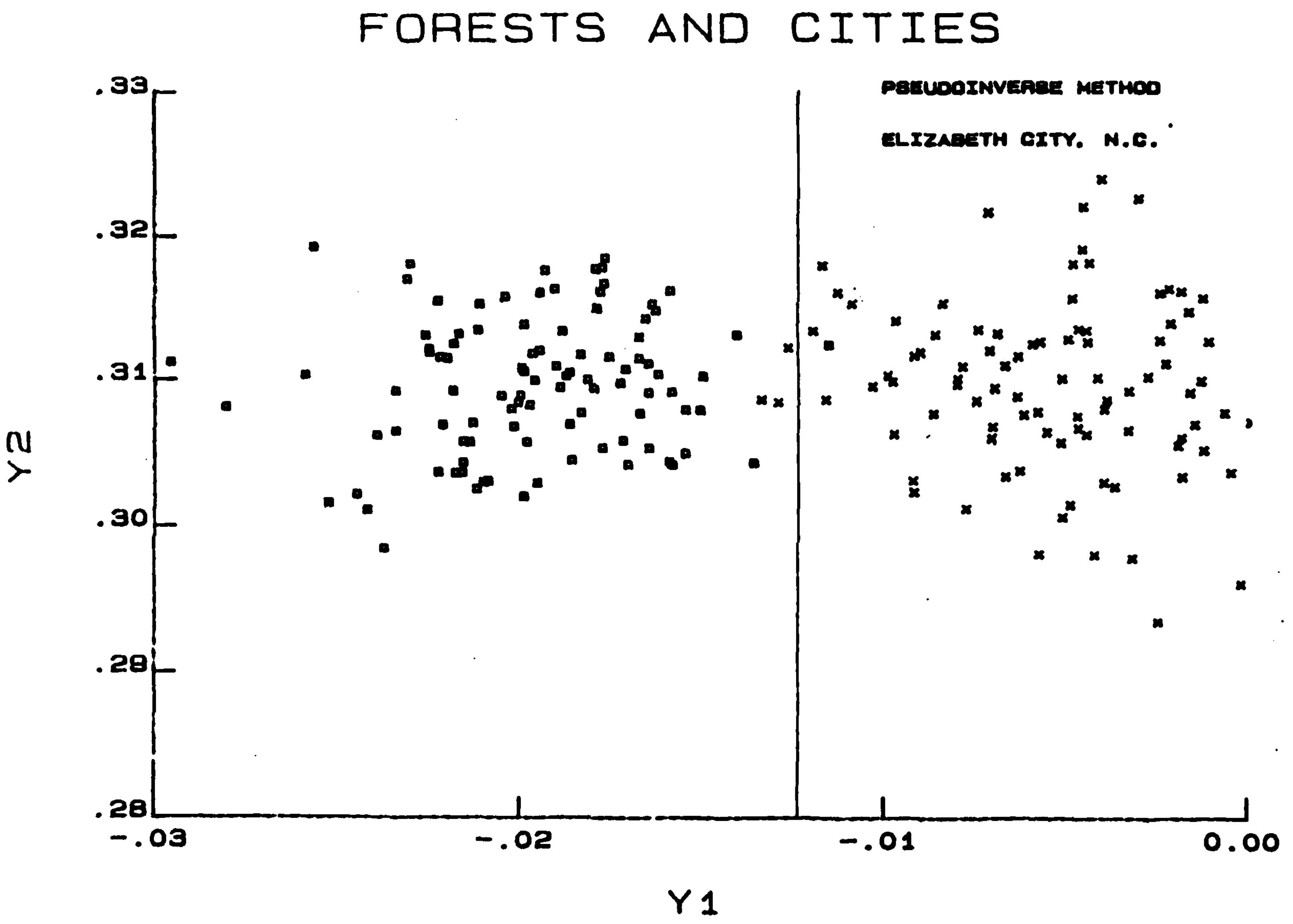


Figure 2. Forests and Cities.

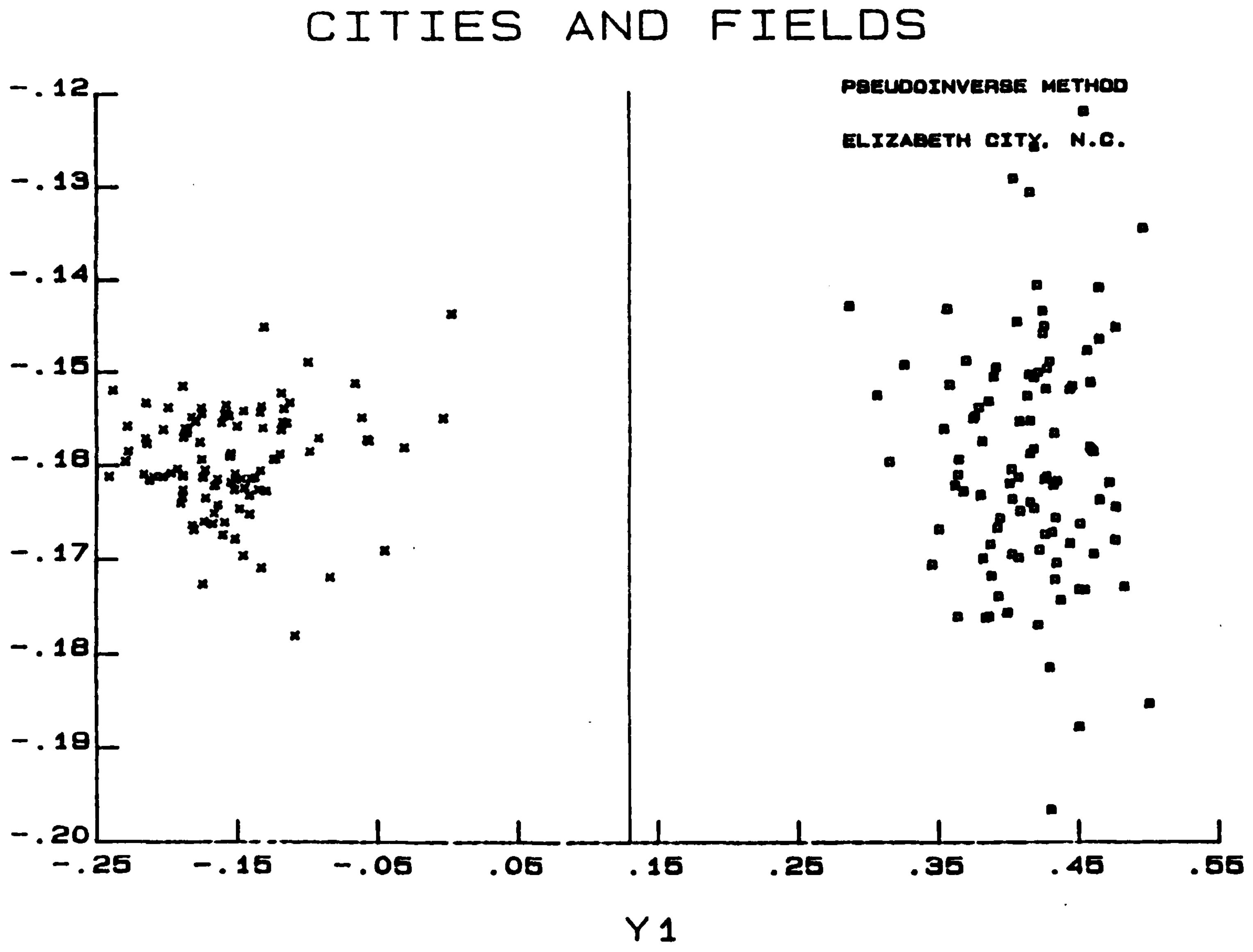


Figure 3. Cities and Fields.

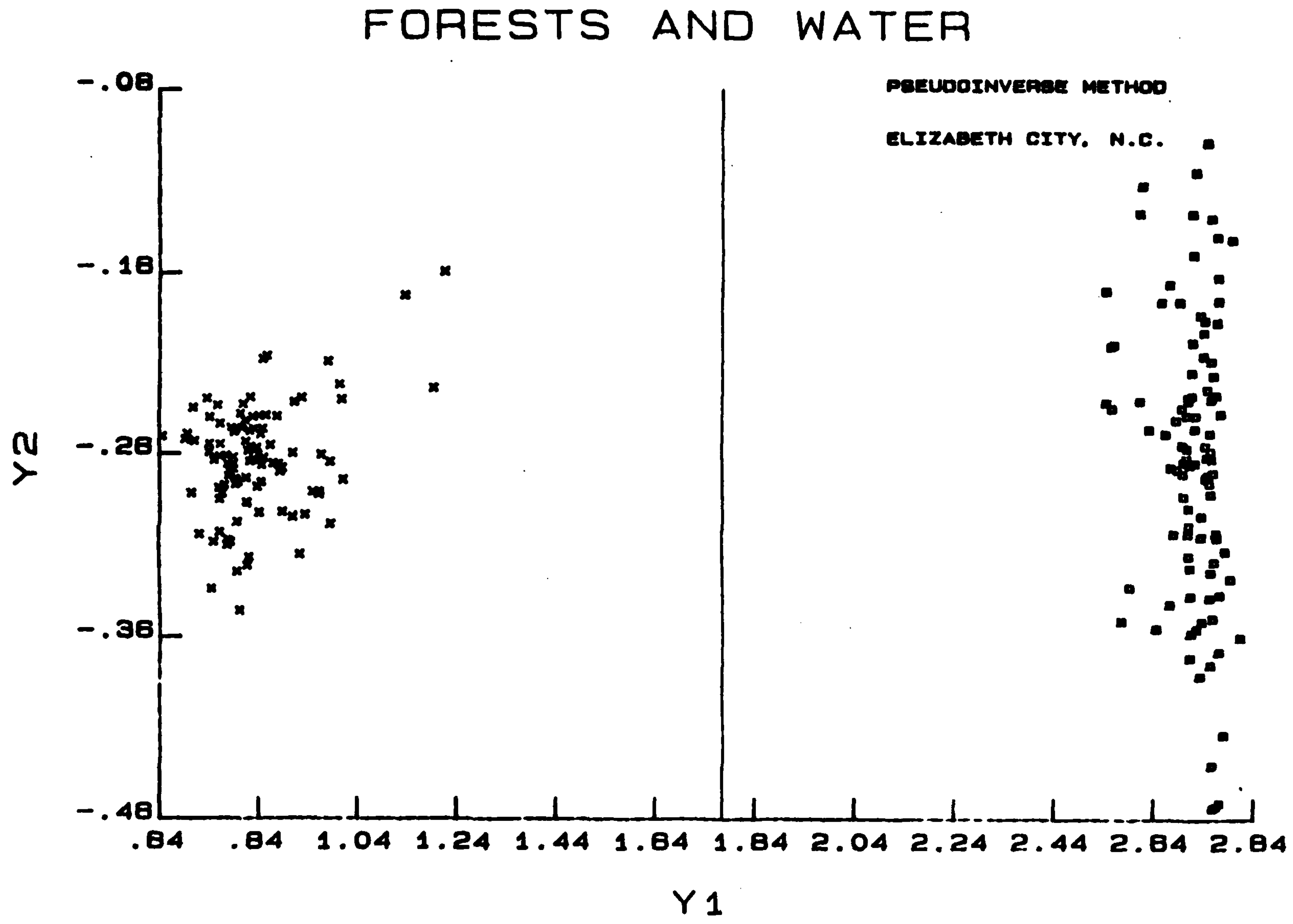


Figure 4. Forests and Water.

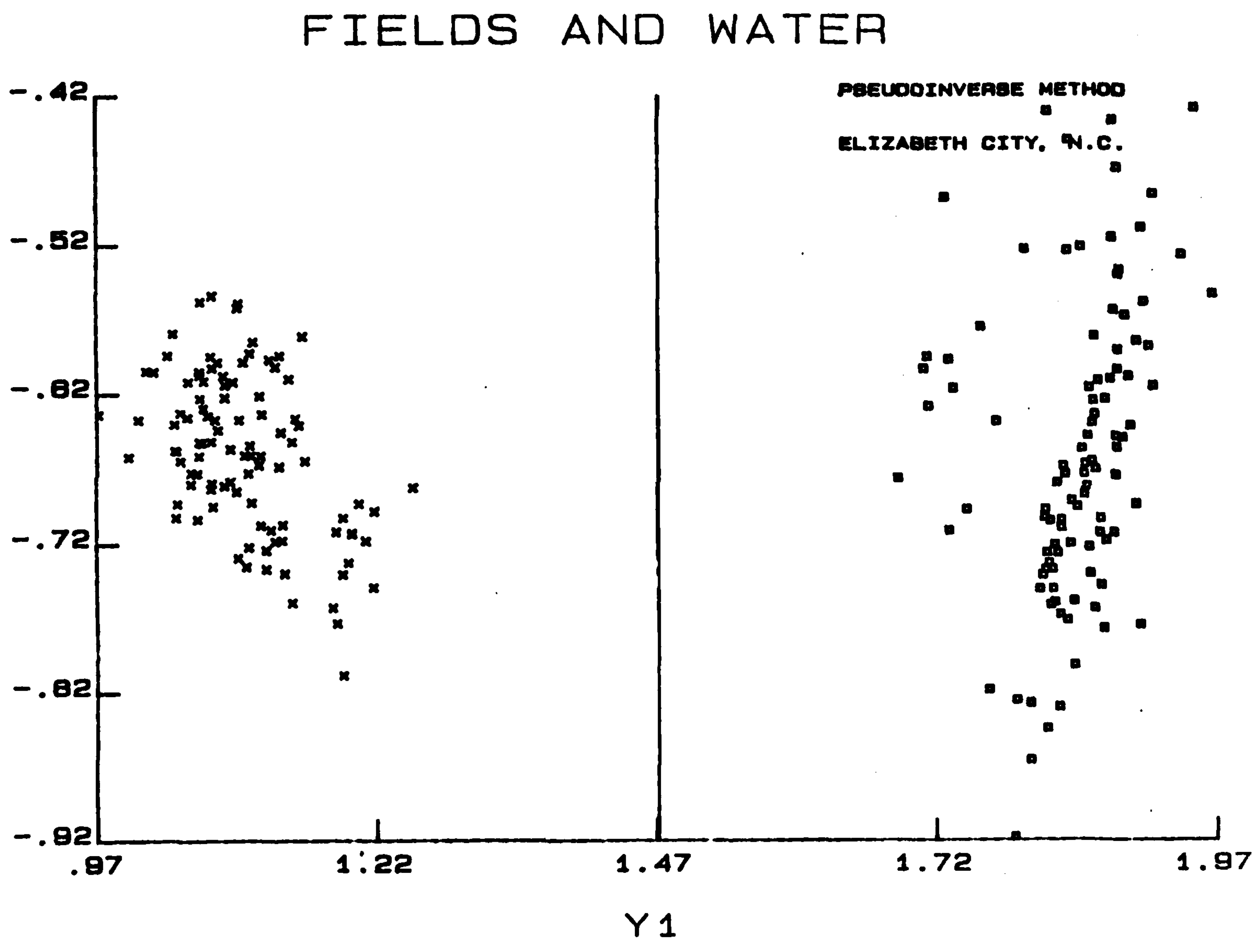


Figure 5. Fields and Water.

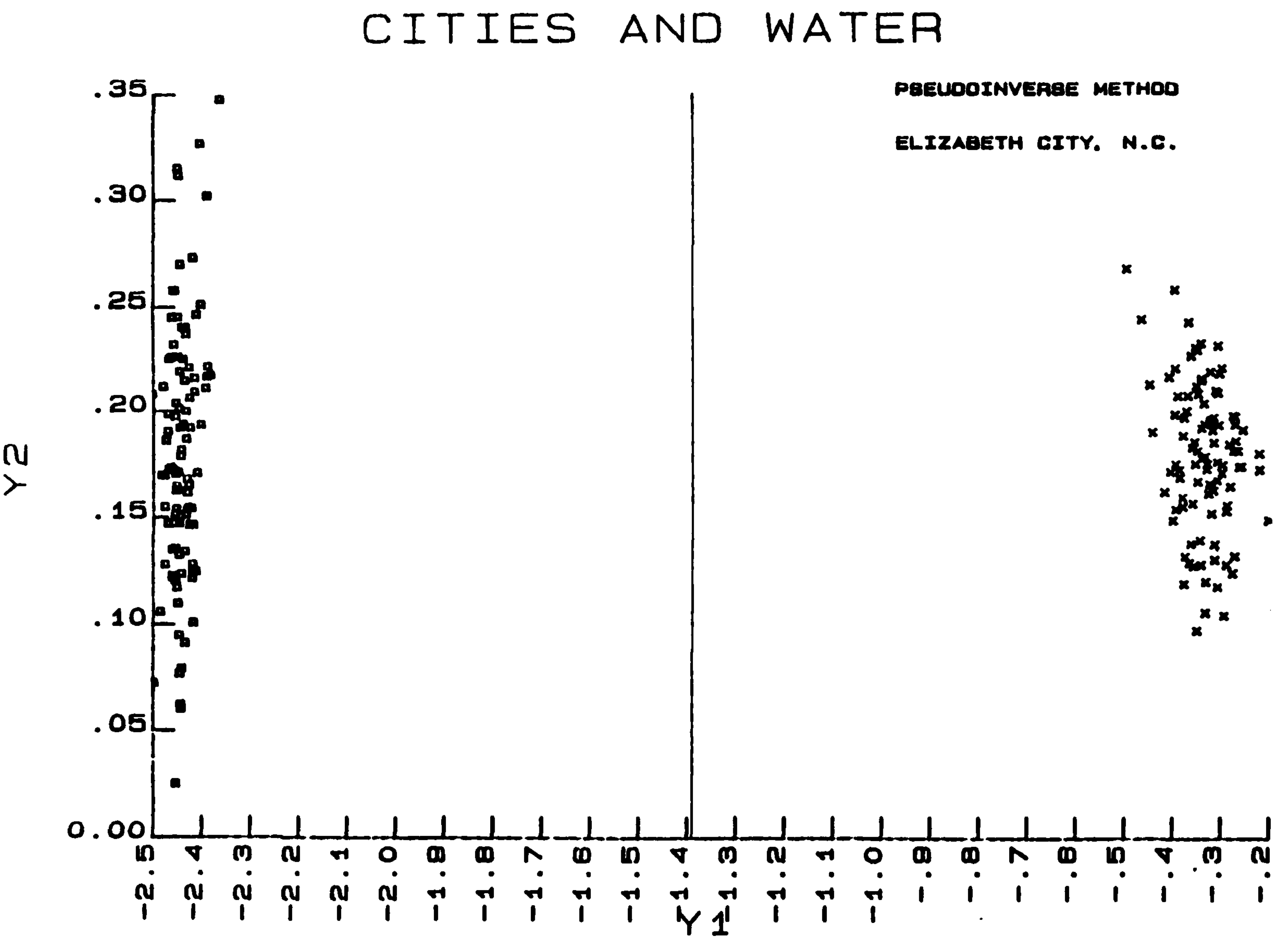


Figure 6. Cities and Water.

Table 2. Results of Classifying tlie Original Training Set

CLASS	NUMBER CORRECT	NUMBER WRONG	PERCENTAGE CORRECT
Forests	94	6	94
Fields	100	0	100
Cities	99	1	99
Water	100	0	100
TOTALS	393	7	98.25%

After the classification technique was shown to work well on the original training set, it was applied to the other 300 samples that came from the same imagery, but were not used to train the classifier. The results of this application are shown in table 3.

Table 3. Results of Classifying the 300 Data Samples

CLASS	NUMBER CORRECT	NUMBER WRONG	PERCENTAGE CORRECT
Forests	93	7	93
Fields	97	3	97
Water	95	5	95
TOTALS	285	15	95%

It can be seen that the third-order approach has worked well on samples taken from the same imagery that was used for training purposes. It remains to be seen if similar results can be obtained with other imagery taken with the same radar system.

CONCLUSIONS

1. The third-order co-occurrence approach for texture analysis was capable of separating four classes of terrain features on radar imagery.

2. The thirteen features calculated from the third-order co-occurrence array provided a 98.25% correct classification rate for the original training set and a 95% correct classification rate for the 300 data samples that were taken in the vicinity of the training set.

3. The excellent correct classification results obtained in this study were also due to the linear feature selection technique used to choose features which were optimized for showing class separability.

Appendix A. Computer Program for Calculating J_1 and the A Matrix along with the Subroutine for Computing the Third-Order Co-occurrence.

```
&SGLNC T=00004 IS ON CR00023 USING 00037 BLKS R=0000

FTN4X,L
C******************************************************************C
C THIS PROGRAM PERFORMS FEATURE SELECTION FOR                      C
C MULTIDISTRIBUTIONS BY MAXIMIZING THE VALUE OF                    C
C J1. THIS PROGRAM USES THE SPATIAL GRAY LEVEL DEPENDENCE TENSOR   C
C******************************************************************C
      PROGRAM SGLNC(3,800),REV.10/13/83
      DIMENSION LUOT(5),IMAGE(1024),X(100,13),IDCB(144),IFIL2(3)
     1,IDYL(35,2),A(13,2),C(13,13,2),S1(13,13),S2(13,13),XMO(13),P(2)
     2,EI(169),LX(13),MX(13),INAM(3),BUF(28),IDXL(35,2),IFIL1(3),
     3 IRES(900),MASK(3,3)
      DATA IANS/2HYE/
      DATA MASK(1,1)/0/,MASK(1,2)/-1/,MASK(1,3)/0/
      DATA MASK(2,1)/-1/,MASK(2,2)/5/,MASK(2,3)/-1/
      DATA MASK(3,1)/0/,MASK(3,2)/-1/,MASK(3,3)/0/
      CALL RMPAR(LUOT)
      LU=LUOT(1)
      CALL ERLU(LU)
    1 NC=2
      NDATA=100
      WRITE(LU,8)
    8 FORMAT("WHICH LINE PRINTER DO YOU WANT TO USE?")
      READ(LU,*)LUP
      WRITE(LU,10)
   10 FORMAT("ENTER THE NAMES OF THE DATA SET FILES")
      READ(LU,17) IFIL1,IFIL2
   17 FORMAT(2(3A2))
      WRITE(LU,15)
      READ(LU,*) IDLU
      WRITE(LU,2104)
 2104 FORMAT("HOW MANY SETS DO YOU WANT TO RUN THROUGH?")
      READ(LU,*) ISET
      DO 99 I=1,ISET
      DO 99 J=1,2
      WRITE(LU,98)J,J,I
   98 FORMAT("ENTER IDX",I1," AND IDY",I1," FOR # ,"I3)
   99 READ(LU,*) IDXL(I,J),IDYL(I,J)
      DO 61 IL=1,ISET
      WRITE(LUP,1330)
 1330 FORMAT(1X,"'A' MATRIX CALCULATIONS FOR THIRD ORDER CO-OCCURRENCE")
      IDX1=IDXL(IL,1)
      IDX2=IDXL(IL,2)
      IDY1=IDYL(IL,1)
      IDY2=IDYL(IL,2)
      WRITE(LUP,1400) IDX1,IDY1,IDX2,IDY2
 1400 FORMAT(2X,"IDX1 = ",I3,"  IDY1 = ",I3/2X,"IDX2 = ",I3,"  IDY2 = "
     1,I3)
      DO 80 NJ=1,NC
   15 FORMAT("DISK LU NUMBER?")
      IF(NJ.EQ.1) THEN
            WRITE(LUP,200) IFIL1
            CALL OPEN(IDCB,IERR,IFIL1,0,0,-IDLU)
            END IF
      IF(NJ.EQ.2) THEN
            WRITE(LUP,200) IFIL2
```

```
           CALL OPEN(IDCB,IERR,IFIL2,0,0,-IDLU)
           END IF
  200 FORMAT(1X,3A2)
      IF(IERR.LT.0) THEN
             WRITE(LU,2010)IERR
 2010        FORMAT("OPEN FILE ERROR",I5)
             GO TO 999 ✓
             END IF
   14 ICONT=1
      CALL LABIN(IDCB,LUP)
   13 J=1
      DO 16 I=1,8
      CALL READF(IDCB,IERR,IMAGE(J))
   16 J=J+128
      IF(IERR.LT.0)GO TO 3000
      GO TO 18
 3000 ICONT=NDATA
      WRITE(LU,2020)IERR
 2020 FORMAT("READ FILE ERROR",I5)
      GO TO  999
   18 NGRAY=16
      CALL LOPER(IMAGE,MASK,IRES,32)
      CALL ISCAL(IRES,IRES,900,0,15)
      CALL SGLDT(IRES,SMO,SNE,ENL,QKN,QIN,QJN,CON,ENT,THO,
     1ABV,A00,A01,A10,NGRAY,IDX1,IDY1,IDX2,IDY2)
C
C======STORE RESULTS IN AN ARRAY=========================C
C
      X(ICONT,1)=SMO
      X(ICONT,2)=SNE
      X(ICONT,3)=ENL
      X(ICONT,4)=QKN
      X(ICONT,5)=QIN
      X(ICONT,6)=QJN
      X(ICONT,7)=CON
      X(ICONT,8)=ENT
      X(ICONT,9)=THO
      X(ICONT,10)=ABV
      X(ICONT,11)=A00
      X(ICONT,12)=A01
      X(ICONT,13)=A10
      IF(ICONT-NDATA)20,22,22
   20 ICONT=ICONT+1
      GO TO 13
   22 CALL CLOSE(IDCB)
      CALL COVER(NDATA,C,A,13,NJ,X,D3)
      WRITE(LUP,70)D3
   70 FORMAT(1X," THE INTRASET DISTANCE=",E15.8)
      WRITE(LUP,140)
  140 FORMAT(2X,"COVARIANCE MATRIX")
      DO 142 K=1,13
      WRITE(LUP,144)(C(K,M,NJ),M=1,13)
  144 FORMAT(1X,13(E9.3,1X))
  142 CONTINUE
      DO 1000 I=1,13
      DO 1000 J=1,13
 1000 S2(I,J)=C(I,J,NJ)
      CALL INV(S2,13,D,LX,MX)
      WRITE(LUP,1010)D
 1010 FORMAT(1X,"THE DETERMINANT OF THE COVARIANCE MATRIX=",E15.8)
```

```
      DO 1020 I=1,13
      DO 1020 J=1,13
 1020 S2(I,J)=0.0
      P(1)=0.5 $ P(2)=0.5
      WRITE(LUP,156)NJ,P(NJ)
  156 FORMAT(1X,"NJ=",I1,5X,"P(NJ)=",F10.8)
   80 CONTINUE
      DO 90 I=1,NC-1
      DO 90 J=I+1,NC
   90 S2(I,J)=0.0
      DO 72 I=1,NC-1
      DO 72 J=I+1,NC
      DO 72 M=1,13
      S2(I,J)=S2(I,J)+(A(M,I)-A(M,J))**2
   72 CONTINUE
      DO 91 I=1,NC-1
      DO 91 J=I+1,NC
      S2(I,J)=SQRT(S2(I,J))
   91 CONTINUE
      WRITE(LUP,73)
   73 FORMAT(1X,"THE INTERSET DISTANCES")
      DO 92  I=1,NC-1
      DO 92  J=I+1,NC
      WRITE(LUP,74)I,J,S2(I,J)
   74 FORMAT(1X,"I=",I1,5X,"J=",I1,5X,"D(I,J)=",E15.8)
   92 CONTINUE
      DO 30 I=1,13
      DO 30 J=1,13
      S2(I,J)=0.0
   30 S1(I,J)=0.0
      DO 32 I=1,13
      DO 32 J=1,13
      DO 32 K=1,NC
   32 S2(I,J)=S2(I,J)+C(I,J,K)*P(K)
      DO 40 J=1,13
      DO 40 M=1,13
   40 S1(J,M)=C(J,M,1)+C(J,M,2)+(A(J,1)-A(J,2))*(A(M,1)-A(M,2))
      DO 41 I=1,13
   41 XMO(I)=0.0
      CALL NROOT(13,S1,S2,XMO,EI)
      DO 165 J=1,13
      DO 165 I=1,13
      K=I+13*(J-1)
      S2(I,J)=EI(K)
  165 CONTINUE
      WRITE(LUP,42)
   42 FORMAT(2X,"EIGENVALUES")
      DO 47 I=1,13
      WRITE(LUP,46)XMO(I)
   46 FORMAT(1X,E15.8)
   47 CONTINUE
      WRITE(LUP,48)
   48 FORMAT(2X,"EIGENVECTORS")
      DO 50 I=1,13
      WRITE(LUP,76)(S2(I,J),J=1,13)
   76 FORMAT(1X,13(F8.5,1X))
   50 CONTINUE
   64 M=2
      XJ1=0.0
      DO 54 I=1,M
   54 XJ1=XJ1+XMO(I)
      WRITE(LUP,56)XJ1,M
   56 FORMAT(2X,"THE VALUE OF J1=",F10.4,5X,"M=",I1)
      CALL TRMAT(S2,S1,13,13,0)
      WRITE(LUP,58)
   58 FORMAT(2X,"THE TRANSFORMATION MATRIX A")
      DO 60 I=1,M
      WRITE(LUP,63)(S1(I,J),J=1,13)
   63 FORMAT(1X,13(E9.3,1X))
   60 CONTINUE
      WRITE(LU,3010)
 3010 FORMAT("DO YOU WANT TO SAVE THE A MATRIX ON A DISK FILE?")
      READ(LU,3001)IAN
 3001 FORMAT(A2)
      IF(IAN.NE.IANS)GO TO 6100
      WRITE(LU,3002)
 3002 FORMAT("INPUT NAME FOR DATA FILE")
      READ(LU,3003)INAM
 3003 FORMAT(3A2)
 3005 FORMAT(I2)
      WRITE(LU,3034)
 3034 FORMAT("INPUT DISK LU #")
      READ(LU,3005)ICR
      CALL CREAT(IDCB,IERR,INAM,1,3,0,-ICR)
      IF(IERR.GE.0)GO TO 3100
      WRITE(LU,3006)IERR
 3006 FORMAT("CREATE FILE ERROR #",I5)
 3100 BUF(1)=FLOAT(M)
      BUF(2)=13.
      IMO=2
      DO 3200 I=1,M
      DO 3200 J=1,13
      IMO=IMO+1
 3200 BUF(IMO)=S1(I,J)
      CALL WRITF(IDCB,IERR,BUF,2*IMO)
      IF(IERR.GE.0)GO TO 3300
      WRITE(LU,3107)IERR
 3107 FORMAT("WRITE FILE ERROR=",I5)
 3300 CALL CLOSE(IDCB,IERR)
      IF(IERR.GE.0)GO TO 4100
      WRITE(LU,3310)IERR
 3310 FORMAT("CLOSE FILE ERROR=",I5)
      GO TO 999
 4100 WRITE(LUP,3320)INAM,ICR
 3320 FORMAT(/,1X,"MATRIX A STORED ON FILE NAMED  ",3A2," LU=",I2)
 6100 WRITE(LUP,96)
   96 FORMAT("1")
   61 CONTINUE
  999 STOP
      END
      SUBROUTINE COVER(KK,COV,AV,N,NJ,X,D3)
      DIMENSION X(100,13),C(13,13),A(13),COV(13,13,2),AV(13,2),VAR(13)
      XK=KK
      DO 50 K=1,N
      A(K)=0.0
      DO 45 J=1,KK
   45 A(K)=A(K)+X(J,K)
   50 A(K)=A(K)/XK
      DO 55 K=1,N
      VAR(K)=0.0
      DO 54 I=1,KK
   54 VAR(K)=VAR(K)+(X(I,K)-A(K))**2
   55 CONTINUE
      DO 60 K=1,N
   60 VAR(K)=VAR(K)/(XK-1.)
      D3=0.0
      DO 65 K=1,N
   65 D3=D3+VAR(K)
      D3=2.*D3
      DO 110 K=1,N
      DO 115 I=1,KK
  115 X(I,K)=X(I,K)-A(K)
  110 CONTINUE
      DO 120 K=1,N
      DO 125 M=1,N
      C(K,M)=0.0
      DO 130 I=1,KK
  130 C(K,M)=C(K,M)+X(I,K)*X(I,M)
      C(K,M)=C(K,M)/XK
  125 CONTINUE
  120 CONTINUE
      DO 135 K=1,N
      DO 135 M=1,N
      AV(K,NJ)=A(K)
  135 COV(K,M,NJ)=C(K,M)
      RETURN
      END
      SUBROUTINE NROOT(M,A,B,XL,X)
      DIMENSION A(169),B(169),XL(13),X(169)
      K=1
      DO 100 J=2,M
      L=M*(J-1)
      DO 100 I=1,J
      L=L+1
      K=K+1
  100 B(K)=B(L)
      MV=0
      CALL EIGEN(B,X,M,MV)
      L=0
      DO 110 J=1,M
      L=L+J
  110 XL(J)=1.0/SQRT(ABS(B(L)))
      K=0
      DO 115 J=1,M
      DO 115 I=1,M
      K=K+1
  115 B(K)=X(K)*XL(J)
      DO 120 I=1,M
      N2=0
      DO 120 J=1,M
      N1=M*(I-1)
      L=M*(J-1)+I
      X(L)=0.0
      DO 120 K=1,M
      N1=N1+1
      N2=N2+1
  120 X(L)=X(L)+B(N1)*A(N2)
      L=0
      DO 130 J=1,M
      DO 130 I=1,J
      N1=I-M
      N2=M*(J-1)
      L=L+1
      A(L)=0.0
      DO 130 K=1,M
      N1=N1+M
      N2=N2+1
  130 A(L)=A(L)+X(N1)*B(N2)
      CALL EIGEN(A,X,M,MV)
      L=0
      DO 140 I=1,M
      L=L+I
  140 XL(I)=A(L)
      DO 150 I=1,M
      N2=0
      DO 150 J=1,M
      N1=I-M
      L=M*(J-1)+I
      A(L)=0.0
      DO 150 K=1,M
      N1=N1+M
      N2=N2+1
  150 A(L)=A(L)+B(N1)*X(N2)
      L=0
      K=0
      DO 180 J=1,M
      SUMV=0.0
      DO 170 I=1,M
      L=L+1
  170 SUMV=SUMV+A(L)*A(L)
  175 SUMV=SQRT(SUMV)
      DO 180 I=1,M
      K=K+1
  180 X(K)=A(K)/SUMV
      RETURN
      END
      END$
```

&DSUBS T=00004 IS ON CR00023 USING 00123 BLKS R=0000

```
      SUBROUTINE SGLDT(IAR,SMO,SNE,ENL,QKN,QIN,QJN,CON,ENT,THO,
     1ABV,A00,A01,A10,NGRAY,IDX1,IDY1,IDX2,IDY2)
      DIMENSION IAR(30,30),COUT(16,16,16)
      SMO=0
      SNE=0
      ENL=0
      QKN=0
      QIN=0
      QJN=0
      CON=0
      ENT=0
      THO=0
      ABV=0
      A00=0
      A01=0
      A10=0
      DO 10 I=1,NGRAY
      DO 10 J=1,NGRAY
      DO 10 K=1,NGRAY
      COUT(I,J,K)=0
   10 CONTINUE
      IF(IDX1.GE.0.AND.IDX2.GE.0)GO TO 12
      IF(IDX1.LT.0.AND.IDX2.GE.0)GO TO 14
      IF(IDX1.GE.0.AND.IDX2.LT.0)GO TO 16
      IF(IDX1.LT.0.AND.IDX2.LT.0)GO TO 18
   12 NXB=1
      IF(IDX1.EQ.0.AND.IDX2.EQ.0)NXE=30
      IF(IDX1.GE.IDX2)NXE=30-IDX1
      IF(IDX2.GT.IDX1)NXE=30-IDX2
      GO TO 20
   14 NXB=1-IDX1
      NXE=30-IDX2
      GO TO 20
   16 NXB=1-IDX2
      NXE=30-IDX1
      GO TO 20
   18 IF(IDX1.LE.IDX2)NXB=1-IDX1
      IF(IDX2.LT.IDX1)NXB=1-IDX2
      NXE=30
   20 CONTINUE
      IF(IDY1.GE.0.AND.IDY2.GE.0)GO TO 22
      IF(IDY1.LT.0.AND.IDY2.GE.0)GO TO 24
      IF(IDY1.GE.0.AND.IDY2.LT.0)GO TO 26
      IF(IDY1.LT.0.AND.IDY2.LT.0)GO TO 28
   22 NYB=1
      IF(IDY1.EQ.0.AND.IDY2.EQ.0)NYE=30
      IF(IDY1.GE.IDY2)NYE=30-IDY1
      IF(IDY2.GT.IDY1)NYE=30-IDY2
      GO TO 30
   24 NYB=1-IDY1
      NYE=30-IDY2
      GO TO 30
   26 NYB=1-IDY2
      NYE=30-IDY1
      GO TO 30
   28 IF(IDY1.LE.IDY2)NYB=1-IDY1
      IF(IDY2.LT.IDY1)NYB=1-IDY2
      NYE=30
      DO 32 I=NYB.NYE
      DO 32 J=NXB.NXE
      I1=IAR(I.J)
      I2=IAR(I+IDY1.J+IDX1)
      I3=IAR(I+IDY2.J+IDX2)
      COUT(I1+1.I2+1.I3+1)=COUT(I1+1.I2+1.I3+1)+1.
   32 CONTINUE
      QC=0
      DO 40 I=1.NGRAY
      DO 40 J=1.NGRAY
      DO 40 K=1.NGRAY
      QC=QC+COUT(I,J,K)
   40 CONTINUE
      DO 42 I=1.NGRAY
      DO 42 J=1.NGRAY
      DO 42 K=1.NGRAY
      XI=I
      XJ=J
      XK=K
      SMO=SMO+(COUT(I,J,K))**2
      SNE=SNE+COUT(I,J,K)/(XI**2+XJ**2+XK**2)
      ENL=ENL+(XI**2+XJ**2+XK**2)*COUT(I,J,K)
      CON=CON+((XI-XJ)**2+(XJ-XK)**2+(XI-XK)**2)*COUT(I,J,K)
      IF(COUT(I,J.K).EQ.0.0)GO TO 44
      ENT=ENT+COUT(I,J,K)*ALOG10(COUT(I,J,K))
   44 THO=THO+(COUT(I,J,K))**3
      ABV=ABV+(IABS(I-J)+IABS(J-K)+IABS(I-K))*COUT(I,J,K)
      IF(IABS(I-J).EQ.0.AND.IABS(I-K).EQ.0)A00=A00+COUT(I,J,K)
      IF(IABS(I-J).EQ.0.AND.IABS(I-K).EQ.1)A01=A01+COUT(I,J,K)
      IF(IABS(I-J).EQ.1.AND.IABS(I-K).EQ.0)A10=A10+COUT(I,J,K)
   42 CONTINUE
      DO 50 I=1.NGRAY
      DO 50 J=1.NGRAY
      QI=0
      QJ=0
      QK=0
      DO 52 K=1.NGRAY
      QI=QI+COUT(K.I,J)
      QJ=QJ+COUT(I,K,J)
      QK=QK+COUT(I.J,K)
   52 CONTINUE
      QIN=QIN+QI**2
      QJN=QJN+QJ**2
      QKN=QKN+QK**2
   50 CONTINUE
      SMO=SMO/QC
      SNE=SNE/QC
      ENL=ENL/QC
      ENT=-ENT/QC
      THO=THO/QC
      ABV=ABV/QC
      QIN=QIN/QC
      QKN=QKN/QC
      QJN=QJN/QC
      RETURN
      END
```

Appendix B. Computer Program for Calculating the a Vector using the Pseudoinverse Technique.

```
&YAXEC T=00004 IS ON CR00023 USING 00024 BLKS R=0000

FTN4,L
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  THIS PROGRAM COMPUTES THE TRANSFORMATION                        C
C  Y=AX WHERE A IS COMPUTED FROM ANOTHER                           C
C  PROGRAM AND IS INPUT HERE                                       C
C                                                                  C
C       THIS IS THE PUESDO-INVERSE METHOD                          C
C                                                                  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      PROGRAM YAXEC(3,300)
      DIMENSION LUOT(5),IMAGE(1024),X(100,13),Y(2,100,2),A(2,13)
      DIMENSION IDCB(144),IFILE(3),YH(200,3),YT(3,200),YY(3,3)
      DIMENSION LX(3),MX(3),YC(3,200),B(200),AA(3),INAM(3),DATAA(50)
     1,INBB(3),IDCB3(144),IDCB2(144),MASK(3,3),IRES(900)
      DATA MASK(1,1)/0/,MASK(1,2)/-1/,MASK(1,3)/0/
      DATA MASK(2,1)/-1/,MASK(2,2)/5/,MASK(2,3)/-1/
      DATA MASK(3,1)/0/,MASK(3,2)/-1/,MASK(3,3)/0/
      CALL RMPAR(LUOT)
      LU=LUOT(1)
      CALL ERLU(LU)
      NC=2
    4 FORMAT(I1)
      WRITE(LU,6)
    6 FORMAT("WHAT IS THE NAME OF THE FILE CONTAINING THE 'A' MATRIX?")
      READ(LU,22)INAM
      WRITE(LU,15)
      READ(LU,*)ICR
      CALL OPEN(IDCB2,IERR,INAM,0,0,ICR)
      IF(IERR.GE.0) GOTO 17
      WRITE(LU,2010)IERR
      STOP
   17 CALL READF(IDCB2,IERR,DATAA)
      IF(IERR.GE.0) GOTO 70
      WRITE(LU,2020)IERR
      STOP
   70 IMG=2
      M=INT(DATAA(1))
      DO 3001 I=1,M
      DO 3001 J=1,13
      IMG=IMG+1
 3001 A(I,J)=DATAA(IMG)
      WRITE(6,501)
  501 FORMAT(5X,"SOLUTION FOR THE 'A' VECTOR-PSEUDOINVERSE TECHNIQUE---
     1NORTH CAROLINA IMAGERY",//)
      WRITE(6,79)
   79 FORMAT(1X,"THE TRANSFORMATION MATRIX A")
      DO 78 I=1,M
      WRITE(6,77)(A(I,J),J=1,13)
   77 FORMAT(1X,13(E9.3,1X))
   78 CONTINUE
      DO 34 NJ=1,2
      DO 34 I=1,100
      DO 34 K=1,2
   34 Y(NJ,I,K)=0
      DO 80 NJ=1,NC
      WRITE(LU,16)
   16 FORMAT("ENTER THE NUMBER OF IMAGE SAMPLES TO BE ANALYZED  LE.100")
      READ(LU,18)NDATA
   18 FORMAT(I3)
      WRITE(LU.20)
   20 FORMAT("ENTER THE FILE NAME FOR THE DATA SET")
      READ(LU.22)IFILE
   22 FORMAT(3A2)
      WRITE(LU,15)
   15 FORMAT("DISK LU NUMBER?")
      READ(LU.2101)IDLU
 2101 FORMAT(I2)
      IF(NJ.GT.1) GOTO 2204
      WRITE(LU,2100)
 2100 FORMAT("ENTER VALUES FOR IDX1.IDY1,IDX2,IDY2")
      READ(LU,*)IDX1,IDY1,IDX2,IDY2
 2204 WRITE(6,1400)IDX1,IDY1,IDX2,IDY2
 1400 FORMAT(2X,"IDX1=",I2,5X,"IDY1=",I2,5X,"IDX2=",I2,5X,"IDY2=",I2)
      WRITE(6,200)IFILE
  200 FORMAT(1X,3A2)
      CALL OPEN(IDCB.IERR.IFILE,0,0,-IDLU)
      IF(IERR.LT.0)GO TO 2000
      GO TO 24
 2000 WRITE(LU,2010)IERR
 2010 FORMAT("OPEN FILE ERROR",I5)
      GO TO 999
   24 ICONT=1
      CALL LABIN(IDCB,6)
   13 J=1
      DO 19 I=1,8
      CALL READF(IDCB,IERR,IMAGE(J))
   19 J=J+128
      IF(IERR.LT.0)GO TO 3000
      GO TO 26
 3000 ICONT=NDATA
      WRITE(LU.2020)IERR
 2020 FORMAT("READ FILE ERROR",I5)
      GO TO 999
   26 NGRAY=16
      CALL LOPER(IMAGE.MASK.IRES.32)
      CALL ISCAL(IRES,IRES,900,0.15)
      CALL SGLDT(IRES,SMO,SNE,ENL,QKN,QIN,QJN,CON,ENT,THO,
     1ABV,A00,A01,A10,NGRAY,IDX1,IDY1,IDX2,IDY2)
      X(ICONT,1)=SMO
      X(ICONT,2)=SNE
      X(ICONT,3)=ENL
      X(ICONT,4)=QKN
      X(ICONT,5)=QIN
      X(ICONT,6)=QJN
      X(ICONT,7)=CON
      X(ICONT,8)=ENT
      X(ICONT,9)=THO
      X(ICONT,10)=ABV
      X(ICONT,11)=A00
      X(ICONT,12)=A01
      X(ICONT,13)=A10
      IF(ICONT-NDATA)28,30,30
   28 ICONT=ICONT+1
      GO TO 13
   30 WRITE(LU,32)NJ,IFILE
   32 FORMAT("NJ=",I1,3X,3A2)
      DO 36 I=1,NDATA
      DO 36 K=1,M
      DO 36 MK=1,13
      Y(NJ,I,K)=Y(NJ,I,K)+A(K,MK)*X(I,MK)
   36 CONTINUE
   80 CONTINUE
      DO 82 I=1,NDATA
      YH(I,1)=1
      YH(I,2)=Y(1,I,1)
      YH(I,3)=Y(1,I,2)
   82 CONTINUE
      NT=2*NDATA
      N1=NDATA+1
      DO 84 I=N1,NT
      IQ=I-NDATA
      YH(I,1)=-1
      YH(I,2)=-Y(2,IQ,1)
      YH(I,3)=-Y(2,IQ,2)
   84 CONTINUE
      CALL TRMAT(YH,YT,200,3,0)
      CALL PRMAT(YT,YH,YY,3,200,3)
      CALL INV(YY,3,D,LX,MX)
      WRITE(6,86)D
   86 FORMAT(1X,"THE DETERMINANT OF YY=",E15.8)
      CALL PRMAT(YY,YT,YC,3,3,200)
      DO 88 I=1,200
      B(I)=1
   88 CONTINUE
      DO 90 I=1,3
      AA(I)=0
   90 CONTINUE
      DO 92 I=1,3
      DO 92 J=1,200
      AA(I)=AA(I)+YC(I,J)*B(J)
   92 CONTINUE
      WRITE(6,94)
   94 FORMAT(1X,"THE WEIGHT VECTOR A IS GIVEN BELOW")
      DO 96 I=1,3
      WRITE(6,98)AA(I)
   98 FORMAT(1X,E15.8)
   96 CONTINUE
      WRITE(LU,3010)
 3010 FORMAT("WOULD YOU LIKE TO SAVE THE 'A' VECTOR ON A DISK FILE?")
      READ(LU,14)IANS
   14 FORMAT(A2)
      IF(IANS.NE.2HYE) GOTO 999
      WRITE(LU,3002)
 3002 FORMAT("INPUT A NAME FOR THE FILE")
      READ(LU,22)INBB
      WRITE(LU,15)
      READ(LU,*)ICR
      CALL CREAT(IDCB3,IERR,INBB,1,3,0,-ICR)
      IF(IERR.GE.0) GOTO 1165
      WRITE(LU,3006)IERR
 3006 FORMAT("CREATE FILE ERROR #* ",I4)
      STOP
 1165 CALL WRITF(IDCB3,IERR,AA,6)
      IF(IERR.GE.0) GOTO 3310
      WRITE(LU,3107) IERR
 3107 FORMAT("WRITE FILE ERROR #: ",I4)
 3310 CALL CLOSE(IDCB3)
      WRITE(6,3320)INBB,ICR
 3320 FORMAT(1X,"THE 'A' VECTOR IS STORED ON FILE NAMED ",3A2," [illegible]
     1I2              ")",
  999 STOP
      END
      END$
```

Appendix C. Computer Program for Classifying Samples.

```
&CLSEC T=00004 IS ON CR00023 USING 00020 BLKS R=0000

FTN4.L
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C THIS PROGRAM CLASSIFIES SAMPLES BASED ON THE                       C
C MINIMUM SQUARED ERROR PSEUDINVERSE TECHNIQUE                        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      PROGRAM CLSEC(3.300)
      DIMENSION LUOT(5).IMAGE(1024).X(13).XQ(3).ISTG(2).INAM(9,6),
     1IDCB(144),IDCF(144),IFILE(3),IFIL(3),ATEN(6,2,13),W(6.3)
     2.XT(6),DATAA(50),ISPAC(4,6),IRES(900),MASK(3,3)
      DATA INAM/2HFO,2HRE,2HST,2HS ,2HAN.2HD ,2HFI.2HEL,2HDS
     1.2HFO.2HRE.2HST.2HS ,2HAN,2HD .2HCI.2HTI.2HES
     2,2HCI,2HTI,2HES,2H A,2HND,2H F,2HIE.2HLD,2HS
     3.2HFO,2HRE.2HST,2HS .2HAN,2HD .2HWA.2HTE.2HR
     4.2HFI.2HEL,2HDS,2H A,2HND,2H W,2HAT,2HER.2H
     4.2HCI.2HTI.2HES.2H A.2HND.2H W.2HAT,2HER.2H  /
      DATA ISTG/2HA .2HW /
      DATA MASK(1,1)/0/,MASK(1,2)/-1/,MASK(1,3)/0/
      DATA MASK(2,1)/-1/,MASK(2,2)/5/,MASK(2,3)/-1/
      DATA MASK(3,1)/0/,MASK(3,2)/-1/,MASK(3,3)/0/
      CALL RMPAR(LUOT)
      LU=LUOT(1)
      CALL ERLU(LU)
      ISPAC(1,1)=1 $ ISPAC(2,1)=0 $ ISPAC(3,1)=2 $ ISPAC(4,1)=0
      ISPAC(1,2)=2 $ ISPAC(2,2)=0 $ ISPAC(3,2)=2 $ ISPAC(4,2)=1
      ISPAC(1,3)=1 $ ISPAC(2,3)=0 $ ISPAC(3,3)=2 $ ISPAC(4,3)=0
      ISPAC(1,4)=1 $ ISPAC(2,4)=0 $ ISPAC(3,4)=2 $ ISPAC(4,4)=0
      ISPAC(1,5)=1 $ ISPAC(2,5)=0 $ ISPAC(3,5)=2 $ ISPAC(4,5)=0
      ISPAC(1,6)=1 $ ISPAC(2,6)=0 $ ISPAC(3,6)=2 $ ISPAC(4,6)=0
      DO 110 J=1,2
      DO 110 I=1,6
      WRITE(LU,1) ISTG(J),(INAM(K,I),K=1,9)
    1 FORMAT("ENTER THE FILE NAME CONTAINING THE ",A2,"MATRIX FOR ",9A2)
 1010 READ(LU,16) IFIL
      WRITE(LU,18)
      READ(LU,*)ICR
      CALL OPEN(IDCF,IERR,IFIL,0,0,ICR)
      IF(IERR.GE.0) GOTO 40
      WRITE(LU,2010) IERR
      STOP
   40 CALL READF(IDCF,IERR,DATAA)
      IF(IERR.GE.0) GOTO 60
      WRITE(LU,2020)IERR
      STOP
   60 IF(J.EQ.2) GOTO 66
      M=INT(DATAA(1))
      IMQ=2
      DO 33 IU=1.M
      DO 33 JU=1.13
      IMQ=IMQ+1
   33 ATEN(I.IU,JU)=DATAA(IMQ)
      GOTO 119
   66 DO 35 IU=1.3
   35 W(I.IU)=DATAA(IU)
  119 CALL CLOSE(IDCF)
  110 CONTINUE
    5 WRITE(LU.10)
   10 FORMAT("ENTER THE NUMBER OF IMAGE SAMPLES TO BE ANALYZED.LE.100")
      READ(LU.12)NDATA
   12 FORMAT(I3)
      WRITE(LU,14)
   14 FORMAT("ENTER THE NAME OF THE DATA SET")
      READ(LU,16)IFILE
   16 FORMAT(3A2)
      WRITE(LU.18)
   18 FORMAT("DISK LU NUMBER?")
      READ(LU.2101)IDLU
 2101 FORMAT(I2)
   20 FORMAT(1X,3A2)
      CALL OPEN(IDCB,IERR,IFILE,0,0,-IDLU)
      IF(IERR.GE.0)GO TO 22
 2000 WRITE(LU,2010)IERR
 2010 FORMAT("OPEN FILE ERROR",I5)
      GO TO 999
   22 ICONT=1
      WRITE(6.20)IFILE
      CALL LABIN(IDCB,6)
   13 J=1
      DO 24 I=1,8
      CALL READF(IDCB,IERR,IMAGE(J))
   24 J=J+128
      IF(IERR.GE.0)GO TO 26
 3000 ICONT=NDATA
      WRITE(LU,2020)IERR
 2020 FORMAT("READ FILE ERROR",I5)
      GO TO 999
   26 NGRAY=16
      CALL LOPER(IMAGE.MASK,IRES,32)
      CALL ISCAL(IRES,IRES,900,0,15)
      DO 300 I=1,6
      IDX1=ISPAC(1,I)
      IDY1=ISPAC(2,I)
      IDX2=ISPAC(3,I)
      IDY2=ISPAC(4,I)
      CALL SGLDT(IRES,X(1),X(2),X(3).X(4).X(5).X(6),X(7),X(8),X(9),
     1X(10),X(11),X(12),X(13),NGRAY,IDX1,IDY1,IDX2.IDY2)
   34 Y1=0
      Y2=0
      DO 36 J=1.13
      Y1=Y1+ATEN(I,1,J)*X(J)
      Y2=Y2+ATEN(I.2,J)*X(J)
   36 CONTINUE
      XQ(1)=1
      XQ(2)=Y1
      XQ(3)=Y2
      XT(I)=0
      DO 38 J=1.3
   38 XT(I)=XT(I)+W(I.J)*XQ(J)
  300 CONTINUE
      KONT1=0
      KONT2=0
      KONT3=0
      KONT4=0
      IF(XT(1).GT.0.0)KONT1=KONT1+1
      IF(XT(1).LT.0.0)KONT2=KONT2+1
      IF(XT(2).GT.0.0)KONT1=KONT1+1
      IF(XT(2).LT.0.0)KONT3=KONT3+1
      IF(XT(3).GT.0.0)KONT3=KONT3+1
      IF(XT(3).LT.0.0)KONT2=KONT2+1
      IF(XT(4).GT.0.0)KONT1=KONT1+1
      IF(XT(4).LT.0.0)KONT4=KONT4+1
      IF(XT(5).GT.0.0)KONT2=KONT2+1
      IF(XT(5).LT.0.0)KONT4=KONT4+1
      IF(XT(6).GT.0.0)KONT3=KONT3+1
      IF(XT(6).LT.0.0)KONT4=KONT4+1
      IF(KONT1.GT.KONT2.AND.KONT1.GT.KONT3.AND.KONT1.GT.KONT4)GO TO 72
      IF(KONT2.GT.KONT1.AND.KONT2.GT.KONT3.AND.KONT2.GT.KONT4)GO TO 74
      IF(KONT3.GT.KONT1.AND.KONT3.GT.KONT2.AND.KONT3.GT.KONT4)GO TO 76
      IF(KONT4.GT.KONT1.AND.KONT4.GT.KONT2.AND.KONT4.GT.KONT3)GO TO 78
      WRITE(6,302)
  302 FORMAT(1X,"UNDETERMINED")
      GO TO 90
   72 WRITE(6,80)
   80 FORMAT(1X,"FOREST")
      GO TO 90
   74 WRITE(6,82)
   82 FORMAT(1X,"FIELDS")
      GO TO 90
   76 WRITE(6,84)
   84 FORMAT(1X,"CITY")
      GO TO 90
   78 WRITE(6,86)
   86 FORMAT(1X,"WATER")
   90 IF(ICONT-NDATA)92,94,94
   92 ICONT=ICONT+1
      GO TO 13
   94 CALL CLOSE(IDCB)
      WRITE(LU,995)
  995 FORMAT("DO YOU WANT TO GO THROUGH AGAIN?")
      READ(LU,994)IAS
  994 FORMAT(A2)
      IF(IAS.EQ.2HYE) GOTO 5
  999 STOP
      END
      END$
```